Privacy and Utility in Compressive Statistical Learning

Rémi Gribonval - DANTE Team, LIP, ENS de Lyon
remi.gribonval@inria.fr
Main Contributors & Collaborators

- Anthony Bourrier, Nicolas Keriven, Antoine Chatalic
- Gilles Puy, Nicolas Tremblay, Yann Traonmilin, Clément Elvira
- Patrick Perez, Mike Davies, Gilles Blanchard, Pierre Vandergheynst
- Laurent Jacques, Vincent Schellekens, Florimond Houssiau, Phil Schniter, Evan Byrne, ...
and special thanks to ...

Team PANAMA, IRISA, Rennes, whether this began
and special thanks to ... 

- Team PANAMA, IRISA, Rennes, whether this began.
- Antoine Chatalic (many slides from his defense)
Large-scale learning

Available data
- training collection of feature vectors = point cloud $\mathcal{X}$

Goals
- infer parameters to achieve a certain task
- generalization to future samples with the same probability distribution

Examples
- PCA: principal subspace
- Clustering: centroids
- Dictionary learning: dictionary
- Classification: classifier parameters (e.g. support vectors)
Large-scale learning

\[ X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \]
Large-scale learning

- High feature dimension $d$
- Large collection size $n = \text{“volume”}$
High feature dimension $d$

Large collection size $n = \text{"volume"}$

Challenge: compress $\mathcal{X}$ before learning?
Compressive learning: three routes

$\mathbf{Y} = \mathbf{M}\mathbf{X}$

**random projections** - Johnson Lindenstrauss lemma
see e.g. [Calderbank & al 2009, Reboredo & al 2013]
Compressive learning: three routes

- **dimension reduction**
- **subsampling**
- **sketching**

\[ X = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix} \]

**Nyström method & coresets**

see e.g. [Williams & Seeger 2000, Agarwal & al 2003, Felman 2010]
Compressive learning: three routes

Inspiration: compressive sensing [Foucart & Rauhut 2013]

Connections with: generalized method of moments [Hall 2005]
kernel mean embeddings [Smola & al 2007, Sriperimbudur & al 2010]
Principle of compressive learning
Utility guarantees
Differential-privacy guarantees
Example: clustering MNIST

Handwritten digits

Spectral embedding

Pre-processing

Sketched Clustering

Sketch

k centroids, each of dimension d

\( n = 70,000 \) points
\( d = 10 \) dimension
\( k = 10 \) clusters

\( m \sim kd \) sketch dimension

\( \tilde{z} \)
Moments & kernel mean embeddings

Data distribution \( X \sim p(x) \)

Sketch = vector of \textit{generalized moments}

\[
z = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \approx \mathbb{E} \Phi(X) = \int \Phi(x)p(x)dx
\]

- \textit{nonlinear} in the feature vectors
- \textit{linear} in the distribution \( p(x) \)

finite-dimensional \textbf{Mean Map Embedding}, \textit{[cf Smola & al 2007, Sriperumbudur & al 2010]}

\[
\mathcal{A}(p) := \mathbb{E}_{X \sim p} \Phi(X)
\]
Learning step: solving an inverse problem

- Sketch = (empirical) \textit{moments}
- Learning from a sketch = \textit{moment matching} problem
- Examples
  - PCA
    - sketch = (random) projection of covariance matrix
    - learning = low-rank matrix recovery
Learning step: solving an inverse problem

- **Sketch** = (empirical) moments
- **Learning from a sketch** = moment matching problem

**Examples**
- **PCA**
  - sketch = (random) projection of covariance matrix
  - learning = low-rank matrix recovery

- **k-means clustering:**
  - sketch = random Fourier moments = characteristic function of data distribution
  - learning = looking for centroids and weights such that

\[
(C, \alpha) = \arg \min_{C, \alpha} \left\| \sum_{i=1}^{k} \alpha_i \Phi(c_i) - z \right\|_2.
\]

- Reminiscent of super-resolution / line-spectral estimation
Comparison with traditional learning

Traditional approach

- ideal goal: minimize risk

\[ R^*(p, \theta) := \mathbb{E}_{X \sim p} \ell(X, \theta) \]

- empirical risk minimization

\[ \hat{\theta} \approx \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, \theta) \]

Compressive learning

- sketch the training data

\[ z = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \in \mathbb{R}^m \]

- optimize a surrogate

\[ \tilde{\theta} \approx \arg\min_{\theta} C'(\theta|z) \]
Comparison with traditional learning

- **Traditional approach**
  - ideal goal: minimize risk

  \[ R^*(p, \theta) := \mathbb{E}_{X \sim p} \ell(X, \theta) \]

  - empirical risk minimization

  \[ \hat{\theta} \approx \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, \theta) \]

  - need access to training samples

- **Compressive learning**
  - sketch the training data

  \[ z = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \in \mathbb{R}^m \]

  - optimize a surrogate

  \[ \tilde{\theta} \approx \arg \min_{\theta} C'(\theta|z) \]

- Computationally expensive.
- High energy consumption.
- Sensitive data (e.g. emails, medical data).
- Multiple passes on the data.
Comparison with traditional learning

**Traditional approach**
- ideal goal: minimize risk

\[ R^*(p, \theta) := \mathbb{E}_{X \sim p} \ell(X, \theta) \]

- empirical risk minimization

\[ \hat{\theta} \approx \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, \theta) \]

- need access to training samples

**Compressive learning**
- sketch the training data

\[ z = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \in \mathbb{R}^m \]

- optimize a surrogate

\[ \tilde{\theta} \approx \arg \min_{\theta} C'(\theta | z) \]

- can « forget » training samples
  - complexity independent of \( n \)
  - potential for privacy preservation
  - good surrogate ???
Comparison with traditional learning

**Traditional approach**
- **ideal goal**: minimize risk

\[ R^*(p, \theta) := \mathbb{E}_{X \sim p} \ell(X, \theta) \]

- empirical risk minimization

\[ \hat{\theta} \approx \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, \theta) \]

- need access to training samples

⚠️ Computationally expensive.
⚠️ High energy consumption.
⚠️ Sensitive data (e.g. emails, medical data).
⚠️ Multiple passes on the data.

**Compressive learning**
- sketch the training data

\[ z = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \in \mathbb{R}^m \]

- optimize a surrogate

\[ \tilde{\theta} \approx \arg \min_{\theta} C(\theta | z) \]

- can « forget » training samples
  - complexity independent of n
  - potential for privacy preservation

✅ good surrogate ??
Optimization landscapes

toy example - clustering
Principle of compressive learning
Utility guarantees
Differential-privacy guarantees
Utility guarantees?

Compressive learning
- ex: mixtures of k Gaussians

Compressive sensing
- ex: k-sparse vectors

\[ \mathcal{P} \mathbf{x} \]

Randomized Generalized Moments

\[ \tilde{\mathbf{z}} = \mathcal{A}(\mathcal{P} \mathbf{x}) + \mathbf{e} \]

Is recovery possible?

\[ \mathbf{y} = \mathbf{A} \mathbf{x} + \mathbf{e} \]

Random Matrix

[Gribonval, Blanchard, Keriven, and Traonmilin, 2020]
Utility guarantees?

**Compressive learning**
- ex: mixtures of $k$ Gaussians

**Compressive sensing**
- ex: $k$-sparse vectors

Statistical guarantees = control of excess risk

Key ideas
- **Task-dependent metric** on distributions
- **Low-dim model set** (task-dependent)

[Bottom reference]

R. GRIBONVAL
Virtual MIA’21, January 11 2021
Utility guarantees?

**Compressive learning**
- ex: mixtures of k Gaussians

**Compressive sensing**
- ex: k-sparse vectors

Statistical guarantees = control of excess risk

Key ideas
- Task-dependent metric on distributions
- Low-dim model set (task-dependent)

Key ideas to achieve small sketches:
- RIP for Mean Map Embedding
- Random kernel approximations [Rahimi & Recht 07, Bach 15]
- Covering dim of « secant set »

[Gribonval, Blanchard, Keriven, and Traonmilin, 2020]
Effect of sketch size $m$
(clustering - planted model)
Principle of compressive learning
Utility guarantees
Differential-privacy guarantees
Learning with *limited memory*

**Memory = limited resource**

Compressive Learning:
- Goal = handle large-scale collections
- "enough information" for learning should be captured

**Privacy = desirable target**

Differential privacy
- Goal = learn without memorizing individual information
- "no more information than needed" should be captured
Learning with *limited memory*

- **Memory = limited resource**
  - Compressive Learning:
    - Goal = handle large-scale collections
    - "enough information" for learning should be captured

- **Privacy = desirable target**
  - Differential privacy
    - Goal = learn without memorizing individual information
    - "no more information than needed" should be captured
Example: speaker verification

- **Step 1: train « Universal Background Model »**
  - main goal = representativity of a wide diversity
    - collect *as much speech data as possible from as many speakers as possible*

- **Step 2: speaker-specific model-adaptation**
  - main goal = do not bother the user!
    - use *as little speech data as possible*

- **Deployment:**
  - hypothesis test given
    - speech utterance
    - claimed speaker identity
Example: speaker verification

Step 1: train « Universal Background Model »

- main goal = representativity of a wide diversity
  - collect *as much speech data as possible from as many speakers as possible*
  - decentralized scenario is favorable
Step 1: train « Universal Background Model »

- main goal = representativity of a wide diversity
  - collect *as much speech data as possible from as many speakers as possible*
  - decentralized scenario is favorable

Example: speaker verification
Step 1: train « Universal Background Model »

- **main goal = representativity of a wide diversity**
  - collect *as much speech data as possible from as many speakers as possible*
  - decentralized scenario is favorable

---

**Example: speaker verification**

![Diagram](attachment:image.png)
Example: speaker verification

- **Step 1: train «Universal Background Model»**
  - main goal = representativity of a wide diversity
    - collect *as much speech data as possible from as many speakers as possible*
    - decentralized scenario is favorable

Device 1: Speech Signal → Local pre-processing → Local Sketches → Global Sketch

Device 2: Speech Signal → Local pre-processing → Local Sketches → Global Sketch

Device 3: Speech Signal → Local pre-processing → Local Sketches → Global Sketch

Device 4: Speech Signal → Local pre-processing → Local Sketches → Global Sketch

- Ulrich Mühe in Florian Henckel von Donnersmarck’s The Lives of Others

- Hypothesis test given speech utterance claimed speaker identity

- Example: speaker verification
Example: speaker verification

**Step 1: train « Universal Background Model »**
- **main goal = representativity of a wide diversity**
  - collect *as much speech data as possible from as many speakers as possible*
  - decentralized scenario is favorable

![Diagram of speech processing](image)

Ulrich Mühe in Florian Henckel von Donnersmarck's The Lives of Others
Example: speaker verification

**Step 1: train « Universal Background Model »**
- main goal = representativity of a wide diversity
  - collect *as much speech data as possible from as many speakers as possible*
  - decentralized scenario is favorable

**Step 2: speaker-specific model-adaptation**
- main goal = do not bother the user!
  - use *as little speech data as possible*

Deployment:
- hypothesis test given
- speech utterance
- claimed speaker identity

Example: speaker verification

Ulrich Mühe in Florian Henckel von Donnersmarck's The Lives of Others
Step 1: train « Universal Background Model »
- main goal = representativity of a wide diversity
  - collect *as much speech data as possible from as many speakers as possible*
  - decentralized scenario is favorable

Example: speaker verification
Example: speaker verification

**Step 1: train « Universal Background Model »**
- main goal = representativity of a wide diversity
  - collect *as much speech data as possible from as many speakers as possible*
  - decentralized scenario is favorable

![Diagram of the speaker verification process](diagram.png)

- **Device 1**
- **Device 2**
- **Device 3**
- **Device 4**

Speech Signal → Local pre-processing → Time-Frequency representation → Local Sketches → Global Sketch → Compressed Learning → Universal Background Model (Gaussian Mixture Model)

Ulrich Mühe in Florian Henckel von Donnersmarck's *The Lives of Others*
Private compressive learning

“Natural” privacy of an aggregated estimator:

\[ z = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \]

- role of sketch size
  - sufficiently large for “task-level” information-preservation
  - sufficiently small for “sample-level” information loss?

Guaranteed privacy?
Formal definition

Definition (Approximate differential privacy): The randomized mechanism $M$ achieves $(\varepsilon, \delta)$-differential privacy iff for any set $S$, and neighbor datasets $X \sim Y$:

$$\Pr[M(X) \in S] \leq \exp(\varepsilon) \Pr[M(Y) \in S] + \delta$$

Notes:

- Notation: $(\varepsilon, \delta)$-DP or $\varepsilon$-DP.
- Different neighboring relations can be considered:
  - replacement of one element ("bounded" DP):
  
  $\begin{array}{c|c}
  X & Y \\
  \hline
  0 & 0 \\
  0 & 1 \\
  1 & 0 \\
  1 & 1 \\
  \end{array}$
  $\sim$
  $\begin{array}{c|c}
  X & Y \\
  \hline
  0 & 1 \\
  0 & 0 \\
  1 & 0 \\
  1 & 1 \\
  \end{array}$

  - add/removal of one element ("unbounded" DP):
  
  $\begin{array}{c|c}
  X & Y \\
  \hline
  0 & 0 \\
  0 & 1 \\
  1 & 0 \\
  1 & 1 \\
  \end{array}$
  $\sim$
  $\begin{array}{c|c}
  X & Y \\
  \hline
  0 & 1 \\
  0 & 0 \\
  1 & 0 \\
  1 & 1 \\
  \end{array}$

[Dwork et al 2006, Balle et al 2018, and many more]
Formal definition

Definition (Approximate differential privacy): The randomized mechanism $M$ achieves $(\varepsilon, \delta)$-differential privacy iff for any set $S$, and neighbor datasets $X \sim Y$:

$$\mathbb{P}[M(X) \in S] \leq \exp(\varepsilon) \mathbb{P}[M(Y) \in S] + \delta$$

Notes:

- Notation: $(\varepsilon, \delta)$-DP or $\varepsilon$-DP.
- Different neighboring relations can be considered
  - replacement of one element ("bounded" DP):
  - add/removal of one element ("unbounded" DP):

[Dwork et al 2006, Balle et al 2018, and many more]
Formal definition

Definition (Approximate differential privacy): The randomized mechanism $M$ achieves $(\varepsilon, \delta)$-differential privacy iff for any set $S$, and neighbor datasets $X \sim Y$:

$$\mathbb{P}[M(X) \in S] \leq \exp(\varepsilon) \mathbb{P}[M(Y) \in S] + \delta$$

Notes:
- Notation: $(\varepsilon, \delta)$-DP or $\varepsilon$-DP.
- Different neighboring relations can be considered
  - replacement of one element (“bounded” DP):
  - add/removal of one element (“unbounded” DP):

Plain sketching mechanism $M(X) = \mathbb{z}(X)$ is **not** DP

Can be made DP by adding Laplacian or Gaussian noise

Privacy-level captured by « sensitivity » of the mechanism

[Dwork et al 2006, Balle et al 2018, and many more]
Private sketching mechanism

### Expressions of sensitivities
- Both for Laplacian and Gaussian noise
- Sharpness of these expressions

[Schellekens et al., ICASSP 2019, Differentially Private Compressive k-Means]
[Chatalic et al., Compressive Learning with Privacy Guarantees, preprint 2020]
The NSR: a proxy for utility

Noise-to-signal ratio:

$$\text{NSR} \triangleq \frac{E\|z(X) - \tilde{s}\|_2^2}{\|\tilde{s}\|^2}$$

RSE = error (relative to k-means)
Subsampling

Compute only $r < m$ features of $\Phi$ when sketching.

**Goal 1:** Reduce the computational complexity.
**Goal 2:** Reduce the amount of released information.
Subsampling

Compute only $r < m$ features of $\Phi$ when sketching.

Proposed mechanism (with subsampling)

\[
\begin{align*}
X & \xrightarrow{\text{subsampled } \Phi} \quad \text{sum and rescale} \quad (\text{here } r = 1) \quad \text{divide by } |X| \\
\end{align*}
\]

**Goal 1:** Reduce the computational complexity.
**Goal 2:** Reduce the amount of released information.

Plain subsampled sketching
- is not DP

[Chatalic et al., Compressive Learning with Privacy Guarantees, preprint 2020]
Subsampling

Compute only $r < m$ features of $\Phi$ when sketching.

**Goal 1:** Reduce the computational complexity.
**Goal 2:** Reduce the amount of released information.

- Plain subsampled sketching
  - is not DP
- Noisy subsampled sketching
  - is DP
  - *sharp expressions* for Laplacian noise

[Chatalic et al., Compressive Learning with Privacy Guarantees, preprint 2020]
Some surprises

- **Subsampling cannot improve DP level**
  - may improve other information theoretic privacy measures?

- **With random Fourier features & Laplacian noise**
  - subsampled sketching **does preserve** DP level while reducing complexity
  - utility (NSR) / complexity tradeoff depends on subsampling strategy

---

[Chatalic et al., Compressive Learning with Privacy Guarantees, preprint 2020]
Learning from random moments: the concept

✓ Guaranteed Statistical Learning with limited memory
✓ Differential-privacy guarantees
Summary

✓ Sketching framework

✓ Statistical guarantees
  ➡ compressive PCA
  ➡ compressive k-means
  ➡ compressive GMM
  ➡ key links with kernels

✓ Privacy guarantees
  ➡ noise & subsampling
  ➡ sharp expressions
  ➡ role of NSR
  ➡ some surprises

✓ Dimension reduction

✓ Empirical success

Ongoing challenges:
- guaranteed algorithms to learn from a sketch?
  e.g.: guarantees for continuous OMP
- sketches for other learning tasks?
  e.g.: classification, sparse matrix factorization
- sharp bounds on sketch sizes
- privacy benefits of subsampling
- exploitation of surprises
- Preprints / papers
  - Statistical Guarantees
    - ... for GMM & k-means
  - Differential Privacy
    - ... with subsampling & sharp bounds
  - Application to Compressive k-Means
    - ... and Compressive GMM
  - Algorithms: continuous OMP
    - ... or Approximate Message Passing
  - Sketching with fast random projections

- Tutorial paper
  - SketchMLBox software toolbox

- arxiv.org/abs/1706.07180
- arxiv.org/abs/2004.08085
- hal.inria.fr/hal-020602080
- hal.inria.fr/hal-02496896
- hal.inria.fr/hal-01386077
- hal.inria.fr/hal-01329195
- hal.inria.fr/hal-02049486
- hal.inria.fr/hal-01991231
- hal.inria.fr/hal-01701121
- arxiv.org/abs/2008.01839
- hal.inria.fr/hal-02960718